

**CLAIMS**

1 1. In a cluster of computing nodes having shared access  
2 to one or more volumes of data storage using a parallel  
3 file system, a method for managing the data storage,  
4 comprising:

5 initiating a session of a data management (DM)  
6 application on a first one of the nodes;

7 running a user application on a second one of the  
8 nodes;

9 receiving a request submitted to the parallel file  
10 system by the user application on the second node to  
11 perform a file operation on a file in one of the volumes  
12 of data storage; and

13 sending a DM event message from the second node to  
14 the first node responsive to the request, for processing  
15 by the data management application on the first node.

1 2. A method according to claim 1, wherein initiating  
2 the session comprises initiating the session in  
3 accordance with a data management application programming  
4 interface (DMAPI) of the parallel file system, and  
5 wherein receiving the request comprises processing the  
6 request using the DMAPI.

1 3. A method according to claim 2, and comprising  
2 receiving and processing the event message at the first  
3 node using one or more functions of the DMAPI called by  
4 the data management application.

1 4. A method according to claim 2, wherein sending the  
2 event message comprises sending the message for  
3 processing in accordance with a disposition specified by  
4 the data management application using the DMAPI for

5 association with an event generated by the file  
6 operation.

1 5. A method according to claim 1, and comprising:  
2 receiving a response to the event message from the  
3 data management application on the first node; and  
4 performing the file operation requested by the user  
5 application on the second node subject to the response  
6 from the data management application on the first node.

1 6. A method according to claim 5, wherein receiving the  
2 request comprises submitting the request using a file  
3 operation thread running on the second node, and blocking  
4 the thread until the response to the event message is  
5 received from the first node.

1 7. A method according to claim 5, wherein sending the  
2 event message comprises passing the event message from a  
3 source physical file system (PFS) on the second node to a  
4 session PFS on the first node, and wherein receiving the  
5 response comprises passing a response message from the  
6 session PFS to the source PFS.

1 8. A method according to claim 1, and comprising:  
2 receiving the event message at the first node;  
3 obtaining a data management access right from a  
4 physical file system (PFS) at the first node responsive  
5 to the event message; and  
6 processing the event message using the access right.

1 9. A method according to claim 1, wherein receiving the  
2 request comprises receiving first and second requests of  
3 different types submitted to a physical file system (PFS)  
4 at the second node, and wherein based on the different  
5 request types, sending the event message comprises

6 sending a first event message to the first node  
7 responsive to the first request, and sending a second  
8 event message responsive to the second request to a  
9 further node, on which a further data management  
10 application session has been initiated.

1 10. A method according to claim 9, wherein sending the  
2 first and second event messages comprises:

3 receiving at the second node a specification of  
4 event types and their respective dispositions, the event  
5 types corresponding to the requests to perform the file  
6 operations, and dispositions indicating which of the  
7 event messages should be sent to which of the nodes; and  
8 sending the messages responsive to the specification.

1 11. A method according to claim 1, wherein running the  
2 user application comprises running a first user  
3 application instance on the second node, and running a  
4 further user application instance on a further one of the  
5 nodes, and comprising receiving a further request  
6 submitted to the parallel file system by the further user  
7 application instance to perform a further file operation,  
8 and sending a further event message responsive to the  
9 further request for processing by the data management  
10 application on the first node.

1 12. A method according to claim 11, wherein the further  
2 one of the nodes is the first node.

1 13. A method according to claim 1, wherein initiating  
2 the session of the data management application comprises  
3 initiating a data migration application, so as to free  
4 storage space on at least one of the volumes of data  
5 storage.

1 14. A method according to claim 1, and comprising  
2 choosing one of the nodes to act as a session manager  
3 node, wherein initiating the session comprises sending a  
4 message from the session node to the session manager  
5 node, causing the session manager node to distribute a  
6 specification of events and respective dispositions of  
7 the events for the session among the nodes in the  
8 cluster, and wherein sending the DM event message  
9 comprises sending the message in accordance with the  
10 dispositions.

1 15. A method according to claim 14, wherein one of the  
2 nodes is appointed to serve as a respective file system  
3 manager for each of one or more file systems in the  
4 cluster, and wherein for each of the file systems, the  
5 session manager node sends the specification of the  
6 dispositions applicable to the file system to the  
7 respective file system manager, which sends the  
8 dispositions to all of the nodes in the cluster on which  
9 the file system is mounted.

1 16. A method according to claim 1, wherein sending the  
2 DM event message comprises incorporating in the message a  
3 data field uniquely identifying the second node.

1 17. A method according to claim 1, and comprising  
2 receiving from one of the nodes other than the first one  
3 of the nodes a call for a data management application  
4 programming interface (DMAPI) function in connection with  
5 the session, and performing the function only if it does  
6 not change a state of the session or of an event  
7 associated with the session.

1 18. Computing apparatus, comprising:

2       one or more volumes of data storage, arranged to  
3 store data; and

4       a plurality of computing nodes, linked to access the  
5 volumes of data storage using a parallel file system, and  
6 arranged so as to enable a data management (DM)  
7 application to initiate a data management session on a  
8 first one of the nodes, while allowing a user application  
9 to run on a second one of the nodes, so that when the  
10 user application submits a request to the parallel file  
11 system on the second node to perform a file operation on  
12 a file in one of the volumes of data storage, a DM event  
13 message is sent from the second node to the first node  
14 responsive to the request, for processing by the data  
15 management application on the first node.

1       19. Apparatus according to claim 18, wherein the session  
2 is initiated in accordance with a data management  
3 application programming interface (DMAPI) of the parallel  
4 file system, and wherein the request is processed using  
5 the DMAPI.

1       20. Apparatus according to claim 19, and wherein the  
2 event message is received and processed at the first node  
3 using one or more functions of the DAPI called by the  
4 data management application.

1       21. Apparatus according to claim 19, wherein the event  
2 message is sent for processing in accordance with a  
3 disposition specified by the data management application  
4 using the DAPI for association with an event generated  
5 by the file operation.

1       22. Apparatus according to claim 17, wherein the nodes  
2 are arranged so that the data management application on

3 the first node generates a response to the event message,  
4 and the file operation requested by the user application  
5 is performed on the second node subject to the response  
6 from the data management application on the first node.

1 23. Apparatus according to claim 22, wherein the request  
2 is submitted using a file operation thread running on the  
3 second node, and the thread is blocked until the response  
4 to the event message is received from the first node.

1 24. Apparatus according to claim 22, wherein the event  
2 message is passed from a source physical file system  
3 (PFS) on the second node to a session PFS on the first  
4 node, and wherein the response comprises a response  
5 message passed from the session PFS to the source PFS.

1 25. Apparatus according to claim 18, wherein when the  
2 event message is received at the first node, a data  
3 management access right is obtained from the physical  
4 file system (PFS) at the first node responsive to the  
5 event message, and the event message is processed using  
6 the access permission.

1 26. Apparatus according to claim 17, wherein when first  
2 and second file operation requests of different types are  
3 submitted to the physical file system (PFS) at the second  
4 node, and wherein based on the different request types,  
5 the second node is arranged to send a first event message  
6 to the first node responsive to the first request, and a  
7 second event message responsive to the second request to  
8 a further node, on which a further data management  
9 application session has been initiated.

1 27. Apparatus according to claim 26, wherein the first  
2 and second event messages are sent after receiving at the

3 second node a specification of event types and their  
4 respective dispositions, the event types corresponding to  
5 the requests to perform the file operations, and  
6 dispositions indicating which of the event messages  
7 should be sent to which of the nodes, such that the  
8 second node sends the messages responsive to the  
9 specification.

1 28. Apparatus according to claim 18, wherein the user  
2 application comprises a first user application instance  
3 running on the second node, and a further user  
4 application instance running on a further one of the  
5 nodes, wherein responsive to a further request submitted  
6 to the parallel file system by the further user  
7 application instance to perform a further file operation,  
8 a further event message responsive to the further request  
9 is sent for processing by the data management application  
10 on the first node.

1 29. Apparatus according to claim 28, wherein the further  
2 one of the nodes is the first node.

1 30. Apparatus according to claim 18, wherein the data  
2 management application comprises a data migration  
3 application, for freeing storage space on at least one of  
4 the volumes of data storage.

1 31. Apparatus according to claim 18, wherein one of the  
2 nodes is chosen to act as a session manager node, wherein  
3 the session is initiated by sending a message from the  
4 first node to the session manager node, causing the  
5 session manager node to distribute a specification of  
6 events and respective dispositions of the events for the  
7 session among the nodes in the cluster, and wherein the

8 DM event message is sent in accordance with the  
9 dispositions.

1 32. Apparatus according to claim 18, wherein one of the  
2 nodes is appointed to serve as a respective file system  
3 manager for each of one or more file systems in the  
4 cluster, and wherein for each of the file systems, the  
5 session manager node is arranged to send the  
6 specification of the dispositions applicable to the file  
7 system to the respective file system manager, which sends  
8 the dispositions to all of the nodes in the cluster on  
9 which the file system is mounted.

1 33. Apparatus according to claim 18, wherein the second  
2 node is arranged to incorporate in the DM message a data  
3 field uniquely identifying the second node.

1 34. Apparatus according to claim 18, wherein upon  
2 receiving from one of the nodes other than the first one  
3 of the nodes a call for a data management application  
4 programming interface (DMAPI) function in connection with  
5 the session, the nodes are arranged to perform the  
6 function only if it does not change a state of the  
7 session or of an event associated with the session.

1 35. A computer software product for use in a cluster of  
2 computing nodes having shared access to one or more  
3 volumes of data storage using a parallel file system, the  
4 product comprising a computer-readable medium in which  
5 program instructions are stored, which instructions, when  
6 read by the computing nodes, cause a session of a data  
7 management (DM) application to be initiated on a first  
8 one of the nodes, while allowing a user application to  
9 run on a second one of the nodes, and in response to a

10 request submitted to the parallel file system by the user  
11 application on the second node to perform a file  
12 operation on a file in one of the volumes of data  
13 storage, cause the second node to send a DM event message  
14 to the first node, for processing by the data management  
15 application on the first node.

1 36. A product according to claim 35, wherein the product  
2 comprises a data management application programming  
3 interface (DMAPI) of the parallel file system, and  
4 wherein the request is processed using the DMAPI.

1 37. A product according to claim 36, and wherein the  
2 event message is received and processed at the first node  
3 using one or more functions of the DAPI called by the  
4 data management application.

1 38. A product according to claim 36, wherein the event  
2 message is sent for processing in accordance with a  
3 disposition specified by the data management application  
4 using the DAPI for association with an event generated  
5 by the file operation.

1 39. A product according to claim 35, wherein the  
2 instructions cause the data management application on the  
3 first node to generate a response to the event message,  
4 whereupon the file operation requested by the user  
5 application is performed on the second node subject to  
6 the response from the data management application on the  
7 first node.

1 40. A product according to claim 39, wherein the request  
2 is submitted using a file operation thread running on the  
3 second node, and the thread is blocked until the response  
4 to the event message is received from the first node.

1 41. A product according to claim 39, wherein the event  
2 message is passed from a source physical file system  
3 (PFS) on the second node to a session PFS on the first  
4 node, and wherein the response comprises a response  
5 message passed from the session PFS to the source PFS.

1 42. A product according to claim 35, wherein when the  
2 event message is received at the first node, a data  
3 management access right is obtained from the physical  
4 file system (PFS) at the first node responsive to the  
5 event message, and the event message is processed using  
6 the access permission.

1 43. A product according to claim 35, wherein first and  
2 second file operation requests of different types are  
3 submitted to the physical file system (PFS) at the second  
4 node, and wherein based on the different request types,  
5 the instructions cause the second node to send a first  
6 event message to the first node responsive to the first  
7 request, and a second event message responsive to the  
8 second request to a further node, on which a further data  
9 management application session has been initiated.

1 44. A product according to claim 43, wherein the first  
2 and second event messages are sent after receiving at the  
3 second node a specification of event types and their  
4 respective dispositions, the event types corresponding to  
5 the requests to perform the file operations, and  
6 dispositions indicating which of the event messages  
7 should be sent to which of the nodes, such that the  
8 second node sends the messages responsive to the  
9 specification.

1 45. A product according to claim 35, wherein the user  
2 application comprises a first user application instance  
3 running on the second node, and a further user  
4 application instance running on a further one of the  
5 nodes, wherein responsive to a further request submitted  
6 to the parallel file system by the further user  
7 application instance to perform a further file operation,  
8 a further event message responsive to the further request  
9 is sent for processing by the data management application  
10 on the first node.

1 46. A product according to claim 45, wherein the further  
2 one of the nodes is the first node.

1 47. A product according to claim 35, wherein the data  
2 management application comprises a data migration  
3 application, for freeing storage space on at least one of  
4 the volumes of data storage.

1 48. A product according to claim 35, wherein the  
2 instructions cause one of the nodes to be chosen to act  
3 as a session manager node, and wherein the session is  
4 initiated by sending a message from the first node to the  
5 session manager node, causing the session manager node to  
6 distribute a specification of events and respective  
7 dispositions of the events for the session among the  
8 nodes in the cluster, and wherein the DM event message is  
9 sent in accordance with the dispositions.

1 49. A product according to claim 35, wherein one of the  
2 nodes is appointed to serve as a respective file system  
3 manager for each of one or more file systems in the  
4 cluster, and wherein for each of the file systems, the  
5 instructions cause the session manager node to send the

6 specification of the dispositions applicable to the file  
7 system to the respective file system manager, which sends  
8 the dispositions to all of the nodes in the cluster on  
9 which the file system is mounted.

1 50. A product according to claim 35, wherein the  
2 instructions cause the second node to incorporate in the  
3 DM message a data field uniquely identifying the second  
4 node.

1 51. A product according to claim 35, wherein upon  
2 receiving from one of the nodes other than the first one  
3 of the nodes a call for a data management application  
4 programming interface (DMAPI) function in connection with  
5 the session, the instructions cause the nodes to perform  
6 the function only if it does not change a state of the  
7 session or of an event associated with the session.